CLAIMS

What is claimed is:

1. A method for running a tamper-resistant application in a trusted environment, comprising:

defining a trusted virtual machine environment that contains a trusted dictionary for protecting data;

verifying the integrity of the application;

wherein, if the application is tampered with, the trusted virtual machine environment prevents the application from accessing secrets in the trusted dictionary, thus disabling the normal operation of the application.

2. The method of claim 1, wherein if the integrity of the application is confirmed, the trusted virtual machine environment allows the application to access the secrets in the trusted dictionary, thus enabling the normal operation of the application.

3. The method of claim 2, wherein defining the trusted virtual machine environment comprises defining a trusted bundle for protecting a programming code of the application.

4. The method of claim 3, wherein protecting the programming code comprises encrypting the programming code.

5. The method of claim 4, wherein the trusted virtual machine environment decrypts the encrypted programming code using a decryption key from a media key block associated with the application.

6. The method of claim 1, wherein defining the trusted virtual machine environment comprises using a security chip.

7. The method of claim 3, wherein defining the trusted bundle comprises restricting access to instruction codes of the trusted bundle.

8. The method of claim 1, further comprising encrypting the trusted dictionary.

9. The method of claim 1, wherein defining the trusted virtual machine environment comprises defining at least two trusted bundles; and

wherein the trusted dictionary is shared between at least some of the at least two trusted bundles, to maintain communication integrity between the at least two trusted bundles.

10. The method of claim 1, wherein the application comprises a player that plays copy-protected media.

11. The method of claim 10, wherein the trusted dictionary contains one or more decryption keys to decrypt the copy-protected media.

12. A method for producing a tamper-resistant application in a trusted virtual machine environment, comprising:

determining whether any aspect of the application needs to be tamper-resistant;

if any aspect of the application needs to be tamper-resistant, defining one or more trusted bundles to restrict access to a predefined set of application functions;

running the one or more trusted bundles in a non-trusted virtual machine environment to debug the application; and

deploying the application in the trusted virtual machine environment.

13. The method of claim 12, wherein the aspect of the application comprises any one or more of: an integrity of the application, a secret key of the application, secret data, and a secret code.

14. The method of claim 12, wherein determining whether any aspect of the application needs to be tamper-resistant comprises determining whether the application needs to access a trusted bundle in another application running in a trusted virtual machine environment.

15. The method of claim 14, further comprising defining a trusted dictionary to be shared between the trusted bundles from the application and the other application.

16. The method of claim 13, wherein if the aspect is any one of a secret key or secret data, building a tool to generate at least one trusted dictionary with the secret key or the secret data.

17. The method of claim 12, wherein the aspect of the application comprises preventing the application from being copied.

18. The method of claim 17, further comprising designing a registration process to determine if the application has been copied.

19. The method of claim 12, wherein the aspect of the application comprises preventing a user from resetting a count of activities of the application.

20. The method of claim 19, further comprising designing a trusted dictionary to contain the count of activities.

21. A computer program product having instruction codes for running a tamper-resistant application in a trusted environment, comprising:

a first set of instruction codes for defining a trusted virtual machine environment that contains a trusted dictionary for protecting data;

a second set of instruction codes for verifying the integrity of the application;

wherein, if the application is tampered with, the trusted virtual machine environment prevents the application from accessing secrets in the trusted dictionary, thus disabling the normal operation of the application.

22. The computer program product of claim 21, wherein if the integrity of the application is confirmed, the trusted virtual machine environment allows the application to access the secrets in the trusted dictionary, thus enabling the normal operation of the application.

23. The computer program product of claim 22, wherein the first set of instruction codes defines the trusted virtual machine environment by defining a trusted bundle for protecting a programming code of the application.

24. The computer program product of claim 23, wherein the first set of instruction codes protects the programming code by encrypting the programming code.

25. The computer program product of claim 24, wherein the trusted virtual machine environment decrypts the encrypted programming code using a decryption key from a media key block associated with the application.

26. The computer program product of claim 21, wherein the first set of instruction codes defines the trusted virtual machine environment comprises using a security chip.

27. The computer program product of claim 23, wherein the first set of instruction codes defines the trusted bundle by restricting access to the trusted bundle.

28. The computer program product of claim 21, further comprising a third set of instruction codes for encrypting the trusted dictionary.

29. The computer program product of claim 21, wherein the first set of instruction codes defines the trusted virtual machine environment by defining at least two trusted bundles; and

wherein the trusted dictionary is shared between at least some of the at least two trusted bundles, to maintain communication integrity between the at least two trusted bundles.

30. The computer program product of claim 21, wherein the application comprises a player that plays copy-protected media.

31. The computer program product of claim 30, wherein the trusted dictionary contains one or more decryption keys to decrypt the copy-protected media.

32. A computer program product having instruction codes for producing a tamper-resistant application in a trusted virtual machine environment, comprising:

a first set of instruction codes for determining whether any aspect of the application needs to be tamper-resistant;

if any aspect of the application needs to be tamper-resistant, a second set of instruction codes defines one or more trusted bundles to restrict access to a predefined set of application functions;

a third set of instruction codes for running the one or more trusted bundles in a non-trusted virtual machine environment to debug the application; and

a fourth set of instruction codes for deploying the application in the trusted virtual machine environment.

33. The computer program product of claim 32, wherein the aspect of the application comprises any one or more of: an integrity of the application, a secret key of the application, secret data, and a secret code.

34. The computer program product of claim 32, wherein the first set of instruction codes determines whether any aspect of the application needs to be tamper-resistant by determining whether the application needs to access a trusted bundle in another application running in a trusted virtual machine environment.

35. The computer program product of claim 34, further comprising a fifth set of instruction codes for defining a trusted dictionary to be shared between the trusted bundles from the application and the other application.

36. The computer program product of claim 33, wherein if the aspect is any one of a secret key or secret data, a sixth set of instruction codes builds a tool to generate at least one trusted dictionary with the secret key or the secret data.

37. The computer program product of claim 32, wherein the aspect of the application comprises a seventh set of instruction codes for preventing the application from being copied.

38. The computer program product of claim 37, further comprising an eight set of instruction codes for designing a registration process to determine if the application has been copied.

39. The computer program product of claim 32, wherein the aspect of the application comprises a ninth set of instruction codes for preventing a user from resetting a count of activities of the application.

40. The computer program product of claim 39, further comprising a tenth set of instruction codes for designing a trusted dictionary to contain the count of activities.

41. A model for running a tamper-resistant application in a trusted environment, comprising:

a definition of a trusted virtual machine environment that contains a trusted dictionary for protecting data;

a verification of the integrity of the application;

wherein, if the application is tampered with, the trusted virtual machine environment prevents the application from accessing secrets in the trusted dictionary, thus disabling the normal operation of the application.

42. The model of claim 41, wherein if the integrity of the application is confirmed, the trusted virtual machine environment allows the application to access the secrets in the trusted dictionary, thus enabling the normal operation of the application.

43. The model of claim 42, wherein the definition of the trusted virtual machine environment comprises a definition of a trusted bundle for protecting a programming code of the application.

44. The model of claim 43, wherein the protection of the programming code comprises an encryption of the programming code.

45. The model of claim 44, wherein the trusted virtual machine environment decrypts the encrypted programming code using a decryption key from a media key block associated with the application.

46. The model of claim 41, wherein the definition of the trusted virtual machine environment comprises the use of a security chip.

47. The model of claim 43, wherein the definition of the trusted bundle comprises a restriction of access to instruction codes of the trusted bundle.

48. The model of claim 41, further comprising an encryption of the trusted dictionary.

49. The model of claim 41, wherein the definition of the trusted virtual machine environment comprises a definition of at least two trusted bundles; and

wherein the trusted dictionary is shared between at least some of the at least two trusted bundles, to maintain communication integrity between the shared trusted dictionary.

50. The model of claim 41, wherein the application comprises a player that plays copy-protected media.

51. The model of claim 50, wherein the trusted dictionary contains one or more decryption keys to decrypt the copy-protected media.

52. A model for producing a tamper-resistant application in a trusted virtual machine environment, comprising:

a determination of whether any aspect of the application needs to be tamper-resistant;

if any aspect of the application needs to be tamper-resistant, a definition of one or more trusted bundles to restrict access to a predefined set of application functions;

a run of the one or more trusted bundles in a non-trusted virtual machine environment to debug the application; and

a deployment of the application in the trusted virtual machine environment.

53. The model of claim 52, wherein the aspect of the application comprises any one or more of: an integrity of the application, a secret key of the application, secret data, and a secret code.

54. The model of claim 52, wherein the determination of whether any aspect of the application needs to be tamper-resistant comprises a determination of whether the application needs to access a trusted bundle in another application running in a trusted virtual machine environment.

55. The model of claim 54, further comprising a definition of a trusted dictionary to be shared between the trusted bundles from the application and the other application.

56. The model of claim 53, wherein if the aspect is any one of a secret key or secret data, a tool generates at least one trusted dictionary with the secret key or the secret data.

57. The model of claim 52, wherein the aspect of the application comprises a prevention of the application from being copied.

58. The model of claim 57, further comprising a design of a registration process to determine if the application has been copied.

59. The model of claim 52, wherein the aspect of the application comprises a prevention of a user from resetting a count of activities of the application.

60. The model of claim 59, further comprising a design of a trusted dictionary to contain the count of activities.